



THE DESIGN OF A
FINANCIAL MANAGEMENT
DATABASE SYSTEM

THESIS

Edward G. Leszynski, Captain, USAF

AFIT/GCA/LAS/96S-10

19970108 012

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

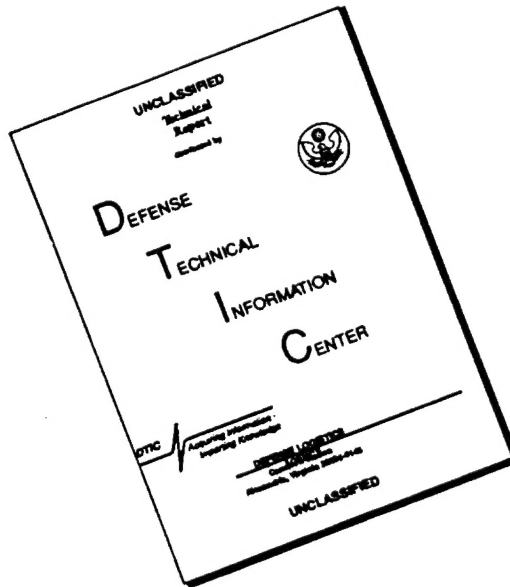
DTIC QUALITY INSPECTED &

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

AFIT/GCA/LAS/96S-10

THE DESIGN OF A
FINANCIAL MANAGEMENT
DATABASE SYSTEM

THESIS

Edward G. Leszynski, Captain, USAF

AFIT/GCA/LAS/96S-10

DTIC QUALITY INSPECTED 3

Approved for public release; distribution unlimited

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

AFIT/GCA/LAS/96S-10

THE DESIGN OF A
FINANCIAL MANAGEMENT
DATABASE SYSTEM

THESIS

Presented to the Faculty of the Graduate School
of Logistics and Acquisition Management
of the Air Force Institute of Technology
Air University

Air Education and Training Command
In Partial Fulfillment of the Requirements for the
Degree of Master of Science in Cost Analysis

Edward G. Leszynski, B.S., M.B.A.

Captain, USAF

September 1996

Approved for public release; distribution unlimited

Acknowledgments

This thesis could not have been accomplished without the help of many individuals. First, I would like to thank Dr. David Christensen, without whom I could never have completed this project. His support and guidance made the thesis task possible. Also, I would like to thank Dr. Norman Ware, who provided much insight and experience to this project. Next, I would like to express my appreciation to my wife, Janice for her encouragement and support. Finally, I would like to thank Major Kyle F. Byard and Mr. Jim Rechterovic for the opportunity to work with them in designing and implementing a state of the art financial management database system.

Edward G. Leszynski

Table of Contents

	Page
Acknowledgments.....	ii
List of Figures.....	v
List of Tables	vi
Abstract.....	vii
 I. Introduction.....	 1
Introduction.....	1
Background	2
General Issues	2
Research Objectives.....	4
Investigative Questions.....	4
Conclusion	5
 II. Literature Review.....	 6
Overview.....	6
Introduction.....	6
Strategies.....	7
Development Methodologies	8
System Development Life Cycle	9
Definition Stage.....	11
Development Stage	17
Recent Studies.....	19
Yadav	20
Hsia	21
Kim	22
El-Rewini	23
Vessey.....	24
Morse.....	26
Lee.....	27
Sharpe.....	27
Conclusion.....	28

	Page
III. Methodology.....	29
Introduction.....	29
Information Requirements.....	29
ASC/FM Requirements.....	30
EM Managers' Requirements.....	31
Feasibility of the System.....	32
Conceptual Design and Prototyping.....	32
Introduction.....	32
Inputs.....	33
Output.....	36
Procedures.....	39
Conclusion.....	40
IV. Results and Analysis.....	41
Introduction.....	41
Research Objectives.....	41
Investigative Questions.....	42
Problems.....	43
Limitations.....	45
Conclusion.....	45
V. Conclusion.....	47
Introduction.....	47
Overview of Investigative Questions.....	47
Recommendations.....	48
Future Research.....	49
Appendix A: Data Dictionary.....	50
Appendix B: The Financial Management Database System User's Manual.....	52
Bibliography.....	63
Vita.....	65

List of Figures

Figure	Page
1. The Receiving Funds Input Screen.....	34
2. The Disbursing Funds Input Screen.....	35
3. Program Financial Statements.....	37
4. Funds Obligation Graphs.....	38
5. Financial Data Menu.....	52
6. EMC Division Financial Data Menu.....	53
7. Checkbook (Program Financial Statements).....	55
8. PMR Chart (Funds Obligation Graphs.....	56
9. Project List.....	57
10. The EMP Data Menu.....	58
11. The Receive Funds Data Input Screen.....	59
12. The Disburse Funds Data Input Screen.....	61

List of Tables

Table	Page
1. The Application System Development Strategies.....	10
2. Summary of Recent Studies.....	20
3. Data Dictionary For User Input Screens.....	50

Abstract

This research led to the design and development of a financial management database system for the Aeronautical Systems Center (ASC) Environmental Management (EM) Systems Program Office (SPO), which has the responsibility of managing the environmental contracts for the Government-Owned, Contractor-Operated (GOCO) plants that are owned by the Air Force. This thesis investigated the various "development strategies" and "methodologies" described in the Management Information Systems literature in order to devise an end-user development strategy capable of meeting the EM SPO's requirements. In addition, the information requirements, conceptual design and prototyping, and procedures phases of the System Development Life Cycle (SDLC) are discussed in detail.

Even though the objective of providing accurate and timely information to program managers was met, problems arose during the design and development phases as a result of changes in requirements. The author thinks that defining information requirements is the most important step in the development process because the ultimate effectiveness of the system depends on how accurately the information requirements are determined initially. Therefore, it is recommended that before a project of this magnitude is undertaken in the future, a firm baseline of system requirements be established early and be adhered to throughout the duration of the project.

THE DESIGN OF A FINANCIAL MANAGEMENT DATABASE SYSTEM

I. Introduction

Introduction

The Aeronautical Systems Center (ASC) Environmental Management (EM) Systems Program Office (SPO) is responsible for managing the environmental contracts for the Government-Owned, Contractor-Operated (GOCO) plants that are owned by the Air Force. Their responsibilities also include collecting and analyzing data for the purposes of evaluating contractor performance, and estimating costs of future requirements. Because these costs have become significant variables in the ultimate decision of retention or transfer of plant ownership, the organization needs to be able to properly identify and assign costs to the various environmental restoration projects at the facilities under their management. In addition, accurate and reliable cost information is required to properly manage current pollution prevention efforts at these plants. However, the Environmental Management SPO does not have in place an accurate and reliable database system. For this reason, ASC/EM management has proposed the

development and implementation of a financial management database system capable of providing accurate and timely information to program managers.

Background

Many Government-Owned, Contractor-Operated (GOCO) plants had their origin during mobilization for WWII when the Army Air Corps acquired approximately one hundred plants. Others were established during the late 1950's and early 1960's to meet the needs for space and ballistic missile programs. It has been government policy to divest itself of ownership of facilities, both real property and plant equipment. Accordingly, since the early 1970's the Air Force has reduced its ownership from thirty to thirteen industrial plants (twelve active and one inactive) which are required for current production of critical weapon systems. These plants, for which Air Force Material Command is landlord, are strategically evaluated annually to determine the need for continued Air Force ownership (Byard). A recent Defense Department environmental report to Congress identified eighty-one military bases and installations with cleanup costs estimated at more than \$100 million each (Bridger, 1995:17). As mentioned above, the ASC EM SPO manages the environmental cleanup contracts for GOCO plants that are owned by the Air Force. The proposed financial management database system will be used in managing these cleanup contracts.

General Issues

Under ASC/FM direction, in December 1995, the Environmental Management (EM) SPO loaded and tested the Integrated Financial Tracking System (IFTS) software,

which had been originally designed specifically for the F-16 SPO. The EM SPO found that their needs were not met. For example, on the various tables in IFTS, money could not be sorted by division and Integrated Product Teams (IPTs) simultaneously. Since the EM SPO consists of four divisions and twelve IPTs which require over fifty financial status reports, this is a major problem. Moreover, funding comes from multiple sources and is allotted to multiple users. The maintenance of fifty separate, but cross-referenced spreadsheets requires an automated database system.

In attempts to identify a system that can allocate monies across different categories, the EM SPO consulted with the developers of the IFTS software at AIRINC and Tecolote. Those developers mentioned that similar problems were occurring at logistics centers in Utah and Sacramento because they, too, receive from various sources monies which are split into different categories. Because the developers had found the problem to be intractable within their existing program structure, they had begun a complete redesign of IFTS. ASC/FM has directed the SPOs to implement IFTS with the hope that the upcoming revision of IFTS will have the desired capability to split funds into different categories. Also, as a result of broad manning cuts, which extend to the Financial Management area, the Environmental Management SPO will not have the manpower to manually keep track of information requirements. For these reasons, it has been decided that the EM SPO should develop its own unique automated financial management database system.

Research Objectives

The objective of this research is to design a user-friendly data base management system for the Environmental Management SPO and a user's manual that will enable the program control office to properly maintain the system. The proposed system must meet the following requirements:

1. The system must remain parallel to IFTS (be compatible, use the same tables as IFTS) because implementation of IFTS may eventually take place. Moreover, because ASC/FM is also a user of the data, and because IFTS utilizes FoxPro software, it was decided that FoxPro would be the software of choice.
2. Due to the planned downsizing of the EM program control office, the system must be completely automated in order to minimize workloads.
3. The system must be user friendly. Managers must be able to easily use the system. In addition, it is essential that the system be capable of simultaneous use by various customers, which will allow managers to stop using their own staffs to maintain separate and often conflicting financial records.

Investigative Questions

Before a database management system can be developed, the following questions must be answered:

1. What are the data reporting requirements of ASC/FM?
2. What information do environmental program managers require to enable them to properly manage their programs?

3. Is it feasible to design within six months a system that will accomplish the objectives of the SPO at no additional cost to the SPO?

Conclusion

A very brief history of GOCO plants and of the responsibilities and challenges of the ASC/EM SPO have been discussed in this chapter in order to familiarize the reader with the reasons behind the decision to design and implement a financial management database system and a supporting user's manual. The next chapter will present a survey of relevant literature and will also explain the design of a database application.

II. Literature Review

Overview

A database is a set of files which store data to support a certain application. Likewise, a database management system (DBMS) is a set of programs which manage this database. The main goal of database systems is to provide managers with information so that they can make effective decisions (Courtney, 1992:7). Because the overall goal of this research is to design a database management system, this literature review will focus on first explaining the basic stages required in developing an application. Next, recent studies and articles that pertain to the various phases will be discussed. Finally, this chapter will review how the findings will affect the design of the financial management database system and the user's manual for the EM SPO.

Introduction

“Development strategy” and “development methodology” are different concepts, and the best strategy for developing an information system application depends on the situation. For each development strategy, there are associated methodologies for performing and managing the process (Davis, 1993:156). A development strategy addresses the amount of trial and error that should be incorporated into development efforts, how much of the development effort should be original work versus building from existing work, and whether parts of the methodology can be simplified or eliminated. Whereas, a development methodology provides a well-defined process by

which an application is conceived, developed and implemented after a strategy is selected (Davis, 1993:157). In other words, the methodology guides the planning and management of the activities involved in developing an application.

Strategies

The four main strategies that help determine how methodologies are applied are: traditional, prototyping, package, and end-user. The traditional strategy incorporates a linear flow of activities through the definition, development, and installation and operation stages. "Underlying the traditional approach are two assumptions: there is a stable set of requirements that can be obtained and documented, and users can verify that the development project is on track from abstract specifications and representations of progress" (Davis, 1993:158).

The prototyping strategy is based on the concept that the eventual user can evaluate an existing system easier than an imagined system. Therefore, it calls for building a working (prototype) system as soon as it is feasible to do so. It emphasizes explicit iterations (rather than a linear flow of activities) based on prototypes that can be tried out by the user. "The prototyping approach overcomes the difficulties of the traditional approach when requirements are not stable or cannot easily be determined because users find it difficult to specify them" (Davis, 1993:158). Also, it works well when users are forced to test the prototype in order to verify that the development project is on track in meeting needs.

"The package approach strategy emphasizes the use of application software packages as the starting point for design and development" (Davis, 1993:158). It is

similar to the prototyping approach due to the existence of the opportunities to try out the system. The difference is that most of the development has already been done.

The end-user development strategy is a simplified approach that is most applicable for small applications and may be applied with either the prototyping or package approach. In creating an end-user strategy for an organization, the user modifies the traditional development strategy in order to meet the specific requirements of the organization. There are many advantages of user-developed systems: such as, it does not require the extensive controls of the traditional approach. For this reason, along with the fact that the Chief of the EM Program Control Office has the experience necessary to develop a database management system, the end-user development strategy was chosen.

Because this strategy has been selected, it is important to point out potential limitations of end-user development. For instance, standards, documentation, controls, testing, and interfaces with other systems, are often neglected, a fact which increases the risk to an organization. In addition, users developing their own systems may underestimate the probability of errors. Moreover, since one person typically does all of the design and development for an application, testing during development has the weakness of the developer testing his (her) own work. For these reasons, the review is a critical issue in quality assurance.

Development Methodologies

Many different types of methodologies exist; however, most are based on one of the following perspectives: process-oriented, data-oriented, and behavior-oriented. The emphasis of the process-oriented perspective is on defining the functions to be performed

and the flow of work. The data-oriented perspective believes that data requirements should be the primary basis for design. The behavior-oriented perspective helps respond to applications that require attention to the order and timing of events. Even though all methodologies must accomplish the same objective, the use of a particular methodology affects the way tasks are defined, the order in which they are accomplished, and the emphasis that is put on some issues.

System Development Life Cycle

The logical process of developing an application from initiation of a project through design and building to implementation, and finally to evolution through corrections is termed a “system development life cycle” (SDLC). A traditional application system development life cycle consists of three major stages: definition, development, and installation and operation (Davis, 1993:160) (see Table 1). The definition stage includes the proposal definition, feasibility assessment, information requirements analysis, and conceptual design. The development stage includes the physical system design, physical database design, program development, and procedure development. The installation and operation stage includes conversion to the new system, operation and maintenance, and post audit.

Table 1 illustrates the similarities and differences among application development strategies. By looking at the table, it is clear that the traditional strategy is the foundation for the other strategies.

TABLE 1
THE APPLICATION SYSTEM DEVELOPMENT STRATEGIES

<u>Traditional</u>	<u>Prototype</u>	<u>Package</u>	<u>End-User</u>
<u>Definition Stage</u>			
Proposal Definition	Proposal	Proposal	
Feasibility Assessment	Feasibility Assessment	Feasibility	
Information Requirements Analysis	Identify Basic Requirements	Request For Proposal (RFP) with "must have" Requirements	Requirements
Conceptual Design and the Use of Data Models			
<u>Development Stage</u>			
Physical System Design			
Database Design	Develop, Use, and Revise until Satisfactory	Identify, Select, and Test a Package	Conceptual Design and Prototyping
Program Development			
Procedure Development	Procedure Development	Procedures	Procedures
<u>Installation & Operation Stage</u>			
Conversion	Conversion	Conversion	
Operation & Maintenance	Operation & Maintenance	Operation & Maintenance	Operation & Maintenance
Post Audit	Post Audit	Post Audit	

Even though the EM SPO's database management system will utilize the end-user development strategy, the various stages in the traditional strategy will be discussed in detail because the other strategies are a deviation of it. For example, the other strategies also have a requirements stage; however, instead of having a separate conceptual design phase like the traditional strategy, the other strategies combine the conceptual design phase with the development stage.

Definition Stage.

Proposal Definition. The first step in the definition stage is the proposal. Proposals may be for entirely new applications or for improvements to existing applications. A proposal provides justification to support a decision to proceed with a feasibility assessment.

Feasibility Assessment. After a new application is proposed, it typically goes through a feasibility study. A feasibility study should at least examine the following five factors: technical, economic, motivational, schedule, and operational. Technical feasibility assessment should address the question of whether or not the proposed application can be implemented with existing technology. Economic feasibility should determine if the system will provide benefits greater than the costs. Motivational feasibility identifies the probability that the organization is sufficiently motivated to support the development and implementation of the application. Schedule feasibility identifies the probability that the organization can complete the development process in the time allowed for development. And finally, operational feasibility assessment examines the likelihood that the system will work when it is installed.

Information Requirements. Once the feasibility study is accepted, information requirements analysis begins. Information requirements analysis is concerned with information as users see it. For example, information is viewed in terms of the way it appears in documents, on terminal screens, or even in images in the user's mind (Courtney, 1992:48). Also, it defines the reports, queries, conceptual schema, and user interface requirements, which are used in subsequent phases to develop the application.

It is widely recognized that determining a complete and correct set of requirements of an organization is vital to the design of an effective information system (Yadav, 1988:1090). The reason for this is that the ultimate effectiveness of the system depends on how accurately the information requirements and user views are specified initially. Several recent studies have evaluated the success of various methods for determining information requirements; however, before discussing these studies, information requirements analysis will be explained in more detail, as will the rest of the System Development Life Cycle.

According to Davis (Davis, 1993:180), information requirements need to be established at the following three levels to enable the successful design and implementation of a computer-based information system:

1. The organizational information requirements to define an overall information system architecture and to specify a portfolio of applications and databases.
2. The requirements for each database defined by data models and other specifications.
3. The detailed information requirements for an application.

Organization-level information requirements determination involves obtaining, organizing, and documenting a complete set of high-level requirements. Also, the overall information architecture is defined, and the boundaries and interfaces of the individual applications are specified (Davis, 1993:181).

Detailed database requirements will include features based on the users' desires. In addition, the requirements for the physical design of the database system will also be defined. User requirements are referred to as conceptual or logical requirements because the user's views of data are separated from the organization of data in physical storage. User requirements may be derived from existing applications or by data modeling (Davis, 1993:181).

There are two types of system application requirements: social and technical. The social or behavioral requirements, which are based on job design, specify objectives and assumptions such as work organization and work design objectives, individual role and responsibility assumptions, and organizational policies. The technical requirements, which are based on the information needed for the job or task to be performed, specify outputs, inputs, stored data, and information processes. Likewise, the technical requirements include interface requirements between the user system and the application system such as data presentation format, screen design, user language structure, feedback and assistance provisions, error control, and response time (Davis, 1993:181).

Even though clear guidelines are given, determining information requirements can be difficult for the following reasons:

1. The variety and complexity of information requirements.

2. The constraints on humans as information processors and problem solvers.
3. The complex patterns of interaction among users and analysts in defining requirements.
4. Unwillingness of some users to provide requirements (for political or behavioral reasons) (Davis, 1993:182).

In order to overcome these challenges four broad strategies for determining information requirements can be used depending on the situation:

1. Asking directly.
2. Deriving from an existing information system.
3. Synthesizing from characteristics of the business processes.
4. Discovering from experimentation with an evolving information system

(Davis, 1993:184).

Conceptual Design. According to Davis, the conceptual design emphasizes the application as seen by the users of the system. It is different than physical design (the next stage) that translates requirements into specifications for implementing the system (Davis, 1993:161). Conceptual design considers the actual processing functions as “black boxes”; whereas physical design defines the actual processing functions.

According to Davis (Davis, 1993:162) the typical contents of a conceptual design include the following:

1. A user-oriented application description that documents the flow of the application activities through the organizational units providing inputs and using outputs and that

distinguishes manual operations from automated operations performed by the application system.

2. Inputs for the application with a general description of each input (such as visual display screens, source documents, forms, and queries).
3. Outputs produced by the application with a general description of each output (such as visual display screens, query responses, printed outputs, and reports).
4. Functions to be performed by the application system.
5. A general flow of processing with relationships of major programs, files, inputs, and outputs.
6. Outlines of operating manuals, user manuals, and training materials needed for the application.
7. Audit and control procedures for ensuring appropriate quality in the use and operation of the application.

Data Models. A data model is a scheme for describing the logical organization of real world entities, the constraints imposed on them and the relationships among them. Logical (or conceptual) data models are used to aid users and designers in specifying data requirements and relationships among data items. The goal of logical data modeling (semantic data modeling) is to accurately and completely represent the data requirements of an application, a function or activity, or an enterprise. The most common approach to logical database specification is the Entity-Relationship (E-R model), which is a graphical method of representing entities, attributes, and relationships (Courtney, 1992:82). Entities are represented by tables in traditional database systems. There are

four models of how a database is organized. The models are hierarchical, network, relational, and object-oriented (Davis, 1993:211).

In a hierarchical model, data are organized hierarchically as a tree structure. A database system based on this model consists of many separate tree structures. Segments, which are tables, are linked in a parent-child relationship. Since all records are stored in an order, every record is accessed by one of three ways: directly, sequentially, or sequentially under current parent. One of the big problems in hierarchical data models is the difficulty in expressing many-to-many relationships. This greatly limits the power of modeling real entities. The ordering of records in a segment makes the insertions and deletions difficult.

By using a network model, some of the problems encountered with a hierarchical model can be avoided. For example, the network structure allows a given entity to have links to related records in other than a top-down approach. Each entity is represented by a table. Many-to-many relationships between two tables can be represented by introducing a third table and defining a set between the third table with each of the other tables. There are three basic ways to access a record: direct, sequential access using a set definition, and simple sequential access. The major disadvantage to the network model is that because users must have explicit knowledge of the relationships represented, it is more complex for a user.

The relational data model is the most popular data model for the traditional database systems that we know today. In the relational model, entities are mapped on to tables. A relation between two tables is also a table. There is no predefined connections

as in the previous two data models. The database is conceived as a set of tables that define simple objects. In each table, the rows represent unique entities or records, and columns represent attributes or data items (Davis, 1993:213). Rather than physical connections built into the database, dynamic connections make the structured modifications of any individual table quite independent.

Whereas in a relational model each table is referred to as a relation, in object-oriented database models, a database object encapsulates data and methods associated with an entity (Davis, 1993:214). Because methods are stored with the entity, the object-oriented database model can be convenient and simple to use, even though it is more difficult to physically implement. Object oriented approaches to systems development are receiving considerable attention in the software engineering arena, and firms, such as Andersen Consulting, are moving toward replacing their current methods with object-oriented ones (Vessey, 1994:102).

Development Stage. The data model produced during the conceptual design phase is combined with the information gathered during the information requirements phase to form the basis for decisions made in the physical design phase of the development stage (Courtney, 1992:81). This stage consists of the physical system design, physical database design, program development, and procedure development.

Physical System Design. According to Davis, (Davis, 1993:162) the results of the physical system design phase are specifications and designs for the following:

- System design showing flow of work, programs, and user functions.

- Control design showing controls to be implemented at various points in the flow of processing.
- Hardware specifications for the applications if new hardware is required.
- Data communications requirements and specifications.
- The overall structure of programs required by the application with procedural specifications on functions to be performed by each.
- Security and backup provisions.
- An application test or quality assurance plan for the remainder of the development.

The aim of the physical system design phase is to take the user requirements of the conceptual design phase and produce a specific technical design. Usually, physical system design techniques achieve simplicity by decomposing the application system into small, relatively self-contained modules. "System complexity is reduced because each module can be developed, coded, and tested independently of the others" (Davis, 1993:164).

Physical Database Design. Physical database design is crucial to successful database system implementation and use. Physical data models describe how to put data items into storage locations so they can be retrieved (Davis, 1993:216). The various types of operations required for a database include: creating the database, locating a record, adding a record, deleting a record, and modifying a record. The approach to physical database design depends on the database requirements and the existing database. Fortunately, the EM SPO should be able to use the data already loaded into the IFTS system enabling this step to be bypassed.

Program Development. The goal of the program development phase is to code and test programs for the application (Davis, 1993:164). Incomplete specifications during the conceptual or physical design phases are the main cause of problems encountered in this phase. Module, integration, and system testing are conducted to establish the level of reliability in the programs. The management of the EM SPO has already decided that all reports will be tested in the attempts of achieving 100% reliability.

Procedure Development. Procedure development (manuals, instruction sheets, input forms, and HELP screens) is important to ensure familiarity for all personnel who have contact with the system. For instance, primary managerial users should have instructions for how to interpret a report and how to select different options for a report. Likewise, secondary data entry users should also be provided instructions for "how to" enter each kind of input. Also, computer operating personnel should have procedures or instructions for quality assurance, backing up system files, and maintaining program documentation. The outcome of this phase will be a user's manual which will enable EM SPO personnel to use and maintain the system.

Recent Studies

Now that the definition and development stages have been explained in detail, the value of recent studies and articles on database design can be evaluated. The following table (Table 2) lists the author's name and the year that the study or article was published, the subject, and a brief summary of the findings.

TABLE 2
SUMMARY OF RECENT STUDIES

<u>Author</u>	<u>Subject</u>	<u>Findings</u>
Yadav (1988)	Comparing Techniques for Information Requirements Analysis	No best technique; provides insight into requirements specification
Hsia (1993)	Requirements Engineering	Object Oriented Techniques still have room for improvement
Kim (1995)	Comparing Data Models	New ideas for information requirements; no model is proven best
El-Rewini (1995)	Object Technology	Using objects for decomposition is more natural than using data functions
Vessey (1994)	Requirements Analysis: Learning Object, Process and Data Methods	Object methods are harder to learn; no one method is best
Morse (1995)	The Evolution of Various Models	Object-Relational Models have emerged as a new model
Lee (1995)	Object-Oriented Models	Introduction of the Object-Relational Model
Sharpe (1995)	Object-Relational Model	Advantages and Description of the Object-Relational Model

Yadav. In his study, Yadav attempts to evaluate the various methodologies for design. Yadav compares the Data Flow Diagram (DFD), the Structured Analysis and Design Technique (SADT), and several other techniques. He reports that although data flow diagrams are easier to learn and use, neither method produces significantly better specifications.

Yadav states that the difficulties in obtaining a complete and correct set of requirements, as were defined by Davis, can be overcome by the use of a systematic method. Specifically, a structured modeling technique can help analysts manage the complexity of an organization by studying each part of it separately while not losing the overall context (Yadav, 1988:1090). Yadav asserts that requirement specification should contain information for the user, designer, implementer, and tester of the system and should include:

1. Functional specification: what functions a system must perform
2. System context, constraints, and assumptions which establish system boundaries
3. Performance specification about the dynamic properties of the system.
4. Measurement and test conditions--an organized testing process to verify that the system is behaving properly (Yadav, 1988:1091).

Hsia. According to Pei Hsia, requirements engineering is one of the most crucial steps in the process of creating quality software products. He defines requirements engineering as the disciplined application of proven principles, methods, tools, and notations to describe a proposed system's intended behavior and its associated constraints (Hsia, 1993:75). It includes all the activities relating to the following:

1. Identification and documentation of customer and user needs.
2. Creation of a document that describes the external behavior and the associated constraints that will satisfy those needs.
3. Analysis and validation of the requirements document to ensure consistency, completeness, and feasibility.

4. Evolution of needs (Hsia, 1993:75).

According to Hsia, problems that hinder the process of requirements engineering are that requirements are difficult to uncover, requirements change, there is an over-reliance on Computer-aided software-engineering tools (CASE tools), training is insufficient, project schedule is always unrealistically tight, developers lack confidence in requirements engineering, communication barriers separate developers and users, and some methods being used are inappropriate.

In the 1970's and early 1980's, many requirements engineers embraced structured analysis. However, Hsia states that this technique, which describes interactions that occur in the real world or among software components, does not allow visualization of the overall system's behavior dynamically and, therefore, tends to lead to premature design (Hsia, 1993:76). Currently, many projects use object-oriented techniques, which provide an effective way to describe entities in the real world and their interactions; however, object-oriented analysis has yet to be demonstrated as effective for documenting a system's external behavior as a "black box". For this reason, before object-oriented methods can be considered a success, a solution to the problem of lack of documentation needs to be addressed.

Kim. According to Young-Gul Kim "accurate specification and validation of information requirements is critical to the development of organizational information systems" (Kim, 1995:103). Kim provides a four-phase process model for requirements determination:

1. Perception-- Users perceive the enterprise reality. The same enterprise reality may be perceived differently by different users (inconsistency). Any one user may perceive only a part of the reality (incompleteness).
2. Discovery--Analysts interact with users to elicit their perceptions.
3. Modeling--Based on the information identified in the discovery phase, analysts build a formal, conceptual model (representation) of the enterprise reality. This model serves as a communication vehicle between analysts and users.
4. Validation-- Before concluding the model is correct, consistent, and complete, it must be validated. Validation has two aspects: comprehension and discrepancy checking. Users must comprehend or understand the meaning of the model. Then they must identify discrepancies between the model and their knowledge of reality (Kim, 1995:103).

Kim states that earlier research identified basically two types of data modeling formalisms: entity-attribute-relationship (EAR) models and object-relationship (OR) models. Moreover, even though proponents of each claim their model yields better representations, Kim concludes that there is little empirical evidence to substantiate these claims (Kim, 1995:103).

El-Rewini. According to Hesham El-Rewini, the procedure-oriented paradigm and structured programming were both popular for a while, and their fundamentals may still be valid. But today, all indicators point to object orientation as a more promising solution (El-Rewini, 1995:58). Unlike the procedural-oriented paradigm, where procedures are the fundamental software building blocks, object-oriented software

consists of interacting objects. Using objects for decomposition is thought to be more natural than using data or functions (El-Rewini, 1995:58).

Vessey. According to Vessey, specifying information requirements is not only the most important step in developing information systems, it is also the most difficult. Moreover, she states that “the crucial aspect of the process is to develop a mental model of the system (i.e., to determine what the system needs to do)”(Vessey, 1994:102). The aid most commonly used in information requirements specification is a systems development methodology and the associated representation technique (Vessey, 1994: 102). Vessey explains that a methodology, is actually a systematic approach to the task of systems development. Also, she says that while early methodologies provided guidelines mostly in the form of a set of standardized activities and standardized forms to be completed, they avoided complexity issues (Vessey, 1994:102). Recently, the major thrust has been to address complexity directly.

Vessey states that humans are limited information processors. Specifying information requirements is extremely difficult for human problem solvers (the second of Davis’s limitations) because it requires handling large amounts of knowledge (Vessey, 1994:102). Such problems are usually handled by decomposing the area under investigation into subproblems for which solutions can be found or generated. In systems development, decomposition is usually achieved by applying methodologies that are designed to formally address ill-structured systems development tasks. In practice, the types of decomposition employed in methodologies employ two bases for decomposition: process and data (Vessey, 1994:103).

Structured analysis is based on the concept of top-down decomposition of systems based on processes. In the conceptual phase, requirements are specified using data flow diagrams, a data dictionary, and process specifications. Data flow diagramming is the technique used to represent the hierarchical decomposition of the system under investigation. Specifically, data flow diagramming achieves top-down partitioning by decomposing the system first into subsystems, then into processes performed within a subsystem; each of those processes is ultimately specified as a processing cycle (Vessey, 1994:103). Consequently, the structured techniques emphasize the processes that transform the data. Data flows are shown as inputs and/or outputs to the subsystems, processes, or steps in the processing cycle. Also, the data dictionary contains definitions of the data in the system (flows and stores). Ultimately, process specifications are developed for the lowest level or primitive processes on the set of data flow diagrams (Vessey, 1994:103).

One data-oriented approach to systems development is termed the “Jackson System Development” (JSD) method. It emphasizes both data and processes, and is conceptually similar to the object-oriented approach. In JSD, requirements are specified using:

1. An entity-action list, which identifies the entities in the systems and the actions that are performed on them, and the actions that they themselves perform.
2. An entity-structure diagram, which shows the order of actions for each entity
3. An initial model (Jackson System Diagram), which connects real-world processes to model processes (Vessey, 1994:103).

Object-oriented analysis and design uses the concept of an object as the unit of decomposition (Vessey, 1994:103). An object is an entity that is characterized by the actions that are imposed on it and the actions it imposes on other objects. The process of object-oriented design interleaves analysis and design of operations relating to objects. For this reason, Vessey states that object-oriented design provides a “more balanced treatment of the objects and operations that exist in the model of the real world than the process approach” (Vessey, 1994:103). The requirements in the object-oriented methodology are specified using a series of lists, which include: the object list, action list, object-attribute list, and the action-attribute list.

Morse. “Twenty years ago, the new gospel of databases hailed the superiority of the emerging relational database model over the older network (Codasyl) model. Relational database management systems (RDBMSs) allow database creation on the fly, or nearly instant changes in a database’s schema that quickly accommodates changes in a corporation’s environment” (Morse, 1995:66). RDBMSs make it easier to create and modify records by storing them on separate tables and linking them through indexes. In contrast, records in the network model are joined through direct and much less mutable relationships (Morse, 1995:66).

On the other hand, object-oriented partisans claim relational databases are ill-suited for storing and manipulating today’s complex data because complexity in a relational database requires creation of more indexes, which can quickly cause a program to slow unacceptably (Morse, 1995:66). “However, a new category has emerged under

the rubric object-relational database and appears to be gaining acceptance among the many database programmers" (Morse, 1995:66).

Lee. Lee explains that in the past, "there have been three approaches to building an Object Database Management System (ODBMS): extending an object-oriented programming language (OOPL), extending a relational DBMS, and starting from the ground up" (Lee, Computer, 1995:64). However, a new paradigm of ODBMS, called object-relational DBMS (ORDBMS), has begun to draw increasing attention. The objective of an ORDBMS is to support both relational and object-oriented database applications. This hybrid results in a database that stores data in a relational form, and also utilizes an object-oriented programming methodology.

Sharpe. Sharpe also refers to this new model that has emerged as an alternative to both relational and object-oriented databases. He states that the object-relational model preserves the fundamental principles of relational theory and still addresses the needs of an object-oriented world (Sharpe, 1995:208DM18). Object-relational databases are similar in philosophy to relational databases because interactions with the database are handled through an enhanced System Query Language (SQL), instead of attempting to match the close level of programming language integration of an object-oriented DBMS. Another advantage of this model is that the object-relational database can store and manipulate objects based on a flexible type system. For this reason, an object can be put in a database without disassembling it, and still take advantage of traditional database features like transactions, query support, and query optimization (Sharpe, 1995:208DM18).

Conclusion

At this time, it is still too early to study or evaluate the effectiveness of the object-relational model. However, Booch, who is known as one of the founding fathers of the object methodology also believes that “it is reasonable to approach the design of a data-centric system by devising a thin object-oriented layer on top of a more traditional relational database technology” (Booch, 1996:127). Booch further states that “this approach is attractive because it leverages the existence of a mature technology (namely, relational databases) yet still permits the core of a system to be cast in object-oriented terms” (Booch, 1996:127).

In conclusion, the System Development Life Cycle for the traditional strategy was explained in detail, even though it has already been decided that the end-user approach to application design is the best approach to meet the needs of the EM SPO. Also, the proper identification of information requirements was stressed as being key to the successful design of an application. In addition, various methodologies or techniques were reviewed as to their usefulness throughout the definition and development phases. Finally, the object-relational model was discussed because it will be utilized along with FoxPro software in developing the financial database management system for the EM SPO. In the next chapter the methodology or process that was followed in the design of the database management system will be explained.

III. Methodology

Introduction

In the last chapter, the traditional strategy of application development was explained in detail even though the end-user development strategy was selected by the EM SPO. In this chapter, the process of developing the financial management database system for the EM SPO is described. In particular, the information requirements, conceptual design and prototyping, and procedures phases of the end-user development strategy as illustrated in Table 1 is described.

Information Requirements

The overall process used to define the original requirements included Major Byard and Mr. Jim Rechterovic, both of the EM SPO, meeting with potential users to discuss the needs of the SPO. Also, I outlined a textbook approach, which was discussed in detail in the previous chapter, that would be most beneficial to the process. In addition, potential areas of concern, as well as an overall evaluation and definition of the requirements of the EM SPO were identified and reviewed by Major Byard, Mr. Rechterovic, and myself.

An initial review of the requirements of the system led to the following (research objectives):

1. Develop a system that remains parallel to IFTS.
2. Create a completely automated system.

3. Create a user-friendly system.

In attempting to properly achieve the research objectives, the following investigative questions were identified:

1. What are the data reporting requirements of ASC/FM?
2. What information do environmental program managers require to enable them to properly manage their programs?
3. Can a system be designed within six months that will accomplish the objectives of the SPO at no additional cost?

ASC/FM Requirements. In addition to remaining parallel to IFTS, it was determined that the system must meet the reporting requirements of ASC/FM. Major Byard, Mr. Rechterovic and myself met with the FM staff on several occasions in order to determine their requirements. After receiving ASC/FM's input, we decided that at a minimum, the system must contain data that are required at quarterly Program Management Reviews (PMRs). This data includes budget authority in all operating years for each appropriation, as well as commitments, obligations, and expenditure data.

In order to ensure that the ASC/FM's requirements were properly identified, definitions were assigned from the Air Force Material Command Financial Management Handbook for the following:

Budget Authority - Authority provided by law to enter into obligations which generally result in the disbursement of Government funds. Also known as obligational authority.

Commitment - A firm administrative reservation of funds for future obligations by local comptrollers. Based on firm procurement directives, orders, requisitions, authorizations

to issue travel orders, or other authorized written evidence which indicate the intent to incur an obligation.

Obligation - A duty to make a future payment of money. The duty is incurred as soon as an order is placed, or contract awarded for the delivery of goods and the performance of services. The placement of an order is sufficient. An obligation legally encumbers a specified sum of money which will require outlays or expenditures in the future.

Expenditures - The final stage of accountability where actual disbursements are made (Air Force Material Command Financial Management Handbook, 1443-1482).

EM Managers' Requirements. Likewise, the needs of ASC/EM managers were defined during a series of round table discussions that were held with FM system developers, Division Chiefs, and Integrated Product Team (IPT) lead managers. Current report formats and data contents were presented and critiqued by the group. Alternate formats were considered, and the group was encouraged to suggest formats and data which would best fit their needs. User requirements were agreed upon, recorded, and used as the basis of the report design.

The information requirements analysis phase determined that ASC/EM managers require continuously updated reports on the financial status of their projects. This status must include the funding requested for the project, the funding received, and funds committed, obligated, and expended. All of this project information must sum to subdivision and division totals, and projects from the various IPTs must be able to sum their data across divisions consistently. Reports on the financial status of IPTs and Division projects must be immediately available to all project managers. Additionally,

graphic displays of this data and comparisons to command goals must be immediately available to all managers for use in briefings and external reports.

Feasibility of the System. Designing the system at no additional cost to the SPO within the six months time was determined to be feasible for several reasons. First, selection of the FoxPro Database software, which was used to develop IFTS, ensured that the compatibility requirement would be met. Second, technical assistance from the developers under contract would be accessible if needed. Third, duplication of development efforts would be reduced. Fourth, the lessons learned from the previous system development effort could be incorporated to save additional development time. Moreover, the Chief of the EM Program Control Office had the experience necessary to develop a database management system, and believed with my help in identifying requirements and documenting the design and development of the system, that six months was a reasonable goal.

Conceptual Design and Prototyping

Introduction. After careful review of the requirements by Major Byard and me, the system was designed by Major Byard using FoxPro software. One advantage of FoxPro software is that its ability to utilize object-oriented programming presents a solution to the problem of successfully managing the many system requirements that the original IFTS system faced, such as dividing monies among various users. Specifically, the use of the project tool enabled object-oriented programming to overlay the traditional relational database system. In other words, the object-relational data model that was

discussed in the previous chapter was applied. The conceptual design was actually accomplished using query maker; however, some code used for the creation of queries, tables and reports was written by copying previous code and changing its parameters.

Inputs. During the conceptual design phase, a change in requirements occurred resulting in the design and creation of two financial management data input screens. The first screen (Figure 1), the Receiving Funds Input screen, displays summary information on all programs and allows the user to enter the receipt of new funds, and to input new projects and funds as they are received by the Directorate. A scrolling table is used to highlight the desired project. Buttons are then used to edit or delete information about that project. The fields available to the user are explained in detail in the user's manual (see Appendix B). The second screen (Figure 2), the Disbursing Funds Input screen, is used to record transactions committing, obligating, and spending those funds, as well as dates when these actions are forecast to occur. The data dictionary containing the parameters of the various fields in the input screens is located in Appendix A.

Microsoft Visual FoxPro 10:41 AM 5/28/98

File Edit View Tools Program Window Help

Receive Funds

Receive Funds

Required	Approved	BA	Released	Committed	SPD Obligated	Obligated	Expenditure
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Project Number: 193577
 Fiscal Year: 1994
 Appropriation: 3010
 BPAC: 140000
 Title: REHAB VERT LIFT DOORS PH/IV
 Description: REHAB VERT LIFT DOORS PH/IV
 OPR Name: EMF
 OCR Name: AFP 04
 Mpc: 3000
 Fund ID:

Funds Required: 594,253.00
 Funds Approved: 708,177.00
 Funds BA: 708,177.00
 Approved Date: / /
 BA Date: / /

Forecast Obligation Amount: 708,177.00
 Forecast Obligation Date: 09/16/96
 Revised Date: / /
 Forecast Expenditure Amount: / /
 Forecast Expenditure Date: / /
 Revised Date: / /

Order By: Project Number Fiscal Year Appropriation OPR Name OCR Name Proj Mtyrappn

Project no	FY	Appn	OPR	OCR	Approved	Committed	Obligated
193577	1994	3010	EMF	AFP 04	708177.00	594253.00	594253.00
193577	1994	3010	EMF	AFP 04	0.00	111924.00	111924.00
193577SC	1994	SALES CR	EMF (SA)	AFP 04	76800.00	76800.00	76800.00
193703	1994	3010	EMC	AFP 42L	171000.00	171000.00	171000.00
193705	1994	3010	EMC	AFP 42L	300000.00	300000.00	299993.00
193993	1994	3010	EMF	AFP 04	74773.00	74773.00	74773.00

Top Prev Next Bottom Find Print Add Edit Delete Cancel Save Exit

Skip to next record NUM CAPS

Figure 1. The Receiving Funds Input Screen

Microsoft Visual FoxPro 3:17 PM 5/30/98

File Edit View Tools Program Window Help

OBLIGAT

Commit, Obligate, Expend

193577 REHAB VERT LIFT DOORS PH IV

Required	Approved	BA	Released	Committed	SPO Obligated	Obligated	Expenditure
594,253.00	706,177.00	706,177.00	706,177.00	706,177.00	706,177.00	706,177.00	0.00

Project Number: 193577
 FY: 1994
 Appn: 3010
 BPAC: 140000
 Document Type: PR
 Document Number: 94-0248
 Contract Number: F33557-97-5-2180
 Contract Mod: F03C33
 Description: REHAB VERT LIFT DOORS PH IV
 Released: 594,253.00
 Release Date: 11/1/94
 Committed: 594,253.00
 Commitment Date: 11/1/94
 SPO Obligated: 594,253.00
 SPO Obligation Date: 09/28/94
 Obligated: 594,253.00
 Obligation Date: 09/28/94

SPO Expenditure: 0.00
 SPO Expenditure Date: 11/1/94
 Expenditure: 0.00
 Expenditure Date: 11/1/94
 Contract Award Date: 05/28/94
 Contractor: LOCKHEED MARTIN
 Msc: 3000
 Fund ID:

Project Number	FY	Appn	OCR	OPR	Document Type	Document Number	Committed	Obligated
193577	1994	3010	AFP	QEMF	PR	94-0248	594,253.00	594,253.00
193577	1994	3010	AFP	QEMF	PR	96-00010	111,924.00	111,924.00
193577SC	1994	SALES	CA	AFP	QEMF	TBD	76,000.00	76,000.00
193703	1994	3010	AFP	4EMC	MIPR	94-5185	171,000.00	171,000.00
193705	1994	3010	AFP	4EMC	PR	94-1008	300,000.00	299,993.00

Top Prev Next Bottom Goto Print Add Edit Delete Cancel Save Exit

Order By: Project Number Appropriation Fiscal Year OPR Name OCR Name

Skip to next record NUM

Figure 2. The Disbursing Funds Input Screen

This newly created database system, known as "Financial Weasel," treats each entry as a discrete transaction. All transactions are recorded in the "Obligat" data table. The summary of these transactions is then recorded in the "Appn" data table. Therefore, each project may appear any number of times in the "Obligat" data table, since any number of transactions, receiving and disbursing funds, may occur for a given program; but a given project number, fiscal year, and appropriation will have only a single entry in "Appn" data table, as this is the "bottom line" for that funding.

Output. The Financial Weasel system will filter, sort, and order the raw data from the transaction tables, group the data for the desired organization, calculate differences and percentages, and display it in the required report. Financial Weasel produces two primary outputs: Program Financial Statements (Figure 3), also known as the Checkbook, and Funds Obligation Graphs (Figure 4), also known as Program Management Review (PMR) charts. Program Financial Statements reflect funds required, approved, received, committed, obligated, expended, and unobligated, as well as percentages of the latter four in relation to the amount approved. These reports are structured as columns of data with rows for each project. As stated earlier, these funds must be divisible by Division, IPT, Appropriation, and Fiscal Year, with summary lines for each. Funds Obligations graphs depict data graphically, with lines for cumulative funds obligated, forecast obligations, revised forecast obligations, and Office of the Secretary of Defense (OSD) obligation goals. Tabular summaries of these figures will also appear on this report.

Report Designer - chkbkopr.frx - Page 2 - Microsoft Visual FoxPro							
File Edit View Tools Program Window Help							
10:57 AM 7/1/98							
EMC 1995							
0701/58 3010							
Project Number	Project Title	MPC	Required	Approved	Committed	Obligated	Expended Unobligated
AFP 42							
13542301	BIO-REMEDIATE FORMER UST SITE PH 22	7000	400,000.00	350,000.00	40,000.00	40,000.00	0.00 310,000.00
Totals			400,000.00	350,000.00	40,000.00	40,000.00	0.00 310,000.00
% of Approved Program					11.43 %	11.43 %	0.00 % 88.57 %
AFP 42L							
3946077	REMOVE UNDERGROUND FUEL LINE BLDG 215	7000	15,469.00	15,469.00	15,469.00	0.00	0.00 15,469.00
33542101	REM & REPL PCB BLDG 211&212-REMEDI SITE 2	7000	0.00	161,306.00	161,306.00	0.00	0.00 161,306.00
Totals			15,469.00	176,775.00	176,775.00	0.00	0.00 176,775.00
% of Approved Program					100.00 %	0.00 %	0.00 % 100.00 %
AFP 42N							
19400319	ASBESTOS & PCB ABATEMENT, B1301	7000	0.00	735,000.00	735,000.00	735,000.00	0.00 0.00
19542408	REMOV HYDRA LIFT/CONTAM ASPHALT BLD	7000	350,000.00	200,400.00	200,400.00	200,400.00	0.00 0.00
19542409	ASBESTOS ABATEMENT, SITE 3	7000	1,000,000.00	256,400.00	256,400.00	256,400.00	0.00 0.00
Totals			1,350,000.00	1,191,800.00	1,191,800.00	1,191,800.00	0.00 0.00
% of Approved Program					100.00 %	100.00 %	0.00 % 0.00 %
AFP 42P							
19542304	CNCT AFP 42P TO LANCASTER REGION WWTP	7000	2,719,000.00	2,719,000.00	2,719,000.00	2,719,000.00	0.00 0.00
33542307	QTRLY GW MONITORING & FIRE TRNG PIT	7000	95,000.00	95,000.00	95,000.00	95,000.00	0.00 0.00
Totals			2,814,000.00	2,814,000.00	2,814,000.00	2,814,000.00	0.00 0.00
% of Approved Program					100.00 %	100.00 %	0.00 % 0.00 %
AFP 42R							
19542201	ASBESTOS ABATEMENT, GDS 1 & 2, SITE 1	7000	350,000.00	350,000.00	350,000.00	0.00	0.00 350,000.00
33542305	RISK MANAGEMENT & PREVENTION PLAN	7000	0.00	30,000.00	30,000.00	30,000.00	0.00 0.00
Totals			350,000.00	380,000.00	380,000.00	30,000.00	0.00 350,000.00
% of Approved Program					100.00 %	7.89 %	0.00 % 92.11 %

Figure 3. Program Financial Statements

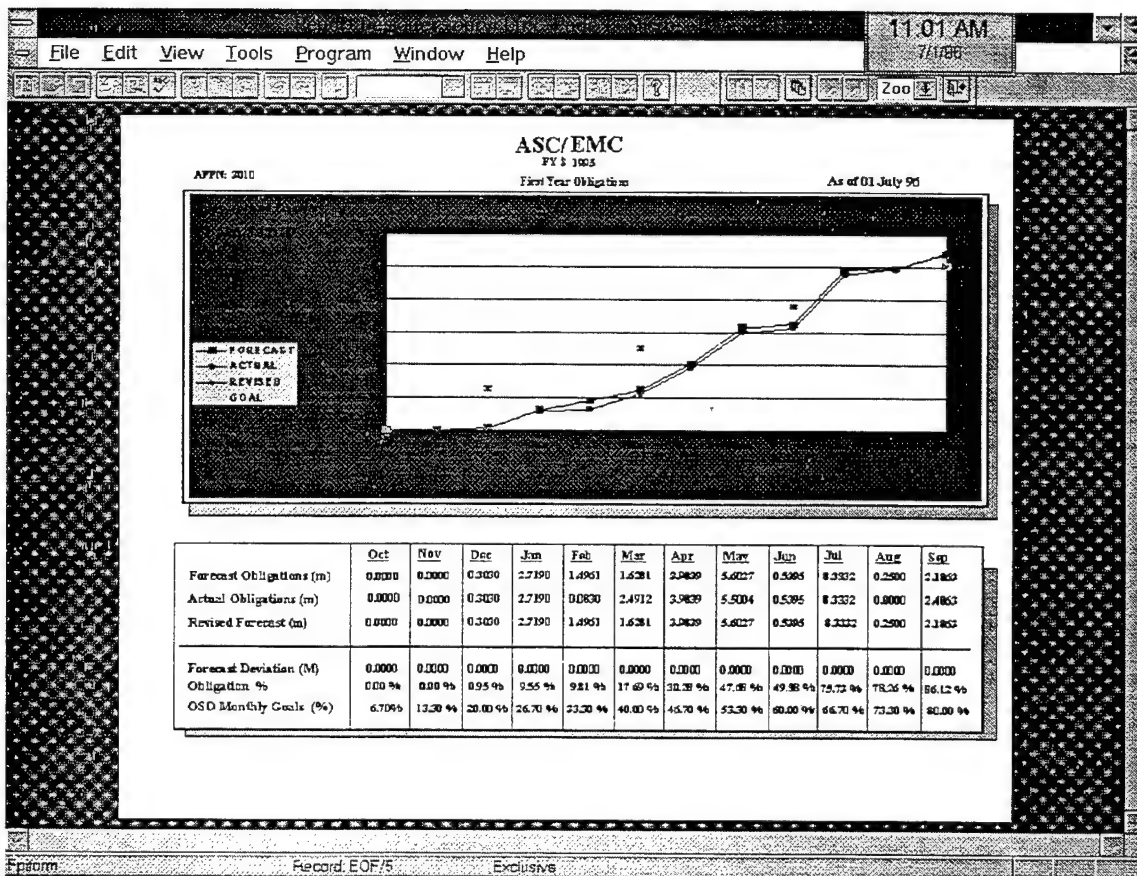


Figure 4. Funds Obligation Graphs

The Financial Weasel system will be self-auditing. Top level reports will include summary lines which can be compared to lower level reports. Because the reports sort the data through several different hierarchies (Division, IPT, appropriation, fiscal year, etc.,) any difference between the lower level reports and the summary lines of the top level report (which includes all data entries) would indicate data not being properly accounted for.

Procedures

Procedure development (manuals, instruction sheets, input forms, and HELP screens) is important to ensure familiarity for all personnel who have contact with the system. For instance, primary managerial users should have instructions on how to interpret a report and how to select different options for a report. Likewise, the data entry users, who will consist of Program Control personnel, should also be provided instructions on how to enter each kind of input. Well documented procedures are essential to insuring proper data entry. Incorrectly entering data, or entering data based upon erroneous definitions, could introduce errors into the database which will corrupt all of the reports which draw off of it.

Also, computer operating personnel should have procedures or instructions for quality assurance, backing up system files, and maintaining program documentation. Documentation of the computer code which generates the reports is also essential. Clearly commented code allows future users of the system to identify the source of problems and to create new applications. The outcome of this phase is a user's manual

designed by myself, which will enable EM SPO personnel to use and maintain the system (See Appendix B).

Conclusion

In this chapter, the process of developing the financial management database system for the EM SPO was described. During the information requirements phase, much effort was expended by Major Byard, Mr. Rechterovic, and myself in attempting to meet all the needs of ASC/FM and the program managers of the EM SPO. As part of the conceptual design phase, Major Byard used the FoxPro software in designing the input and output screens, which were discussed in this chapter. And finally, as part of the procedures phase, I designed the user's manual which is Appendix B. In the next chapter, the results and analysis of this project will be described.

IV. Results and Analysis

Introduction

The Financial Weasel Database Management System was implemented on 1 May 96. The EM SPO program manager was extremely pleased with the system's performance. In this chapter, the original research objectives and investigative questions will be reviewed. In addition, problems that occurred during design and development, as well as limitations of the system will be discussed.

Research Objectives

Originally, the first objective of the system was to remain parallel to the IFTS database. However, after the initial prototype design was created, further requirements were identified which required changing the original design of the system. These changes came after a meeting with the ASC/FM Director, who confirmed that a new IFTS system might never actually be implemented command wide due to the problems and issues raised from users, including the EM SPO. This removed the requirement to remain compatible with IFTS, a requirement which had constrained much of the original design. With this new information at hand, it was decided to create an original database instead of using the IFTS system. Because both databases utilize FoxPro software, no major innovations were required. In fact, the data tables to date were simply transferred to the new database. On the other hand, input screens for the new database had to be

designed and tested, which presented several problems, including what possible combinations of inputs could potential users make that would adversely affect the system.

The second objective of having a completely automated system was successfully met. Likewise, the third objective of creating a user-friendly system was also met. Both of these objectives can be verified by the user's manual (Appendix B), which demonstrates the user-friendly menu screens which enable a user to easily navigate through the system.

Investigative Questions

The first investigative question was to determine the data reporting requirements of ASC/FM. It was determined that the system must contain data that are required at quarterly Program Management Reviews (PMRs). This data includes budget authority in all operating years for each appropriation, as well as commitments, obligations, and expenditure data. The output displayed in the Checkbook screen and the PMR Chart screen clearly demonstrate that this information has been incorporated into the system.

The second investigative question concerning the information requirements of environmental program managers presented the main problem of the information requirements analysis. Even though much time was spent identifying the initial system requirements, this phase, as addressed in the literature review, continued to plague the developers throughout the development process. For instance, after reviewing the initial outputs of the Financial Weasel System, ASC/EM managers desired more and more elaborate products. Each of these additional requirements translated into redesign of parts of the system and the creation of new executable code which had to be integrated

and tested into existing procedures. All of these changes took time, pressing the developers against schedule constraints.

One such example of “requirements creep” concerned the types of graphs and reports the system produced. It seemed that at every review meeting with the users, changes were made in attempts to improve reports. Even though this seems like a logical process, there comes a time when the users have to make a final decision on their needs and wants instead of constantly having the designer switch designs for a “let’s see how this looks” test. However, in the end, the program managers were extremely pleased with the final output.

The third investigative question was to determine the feasibility of designing the system in six months at no extra cost to the SPO. At the completion of the project, the SPO had not incurred any additional costs; however, information requirements changes resulted in changes in the design of the system. These changes in design resulted in the system completion date of mid-June 1996, which slightly exceeded the six-month time-frame established in early December 1995.

Problems

The requirement of permitting multiple users on the system simultaneously was difficult to solve. At first, users were often locked out when code was being written or modified. However, the use of the table buffering in the FoxPro software proved to be the answer. The use of optimistic table buffering was aided by insight provided by ARINC, the original developers of the IFTS system. As was mentioned in the literature review, the use of the object-oriented programming layer, which in this case was the data

buffering scheme, allows a relational database added dimensions. Optimistic table buffering took care of all relationship problems and allowed multiple users.

Optimistic table buffering allows multiple users to make changes to the same table simultaneously by committing these changes to a memory buffer until the user commits them to the permanent table. While a powerful tool, this configuration demanded significant changes in the system code. Optimistic data buffering requires that the data tables be opened in buffered mode and left open while the system is in use. Therefore, code openings and closings had to be changed.

As was mentioned in the literature review, problems can also arise when the developer of an end-user system is also the main tester of the system. For this reason, Mr. Jim Rechterovic was selected to use the prototype system on a daily basis to test for potential problems or glitches, as well as evaluate ease of use and areas that required further development or better design.

For several months, it seemed that Mr. Rechterovic was constantly bringing problems to the attention of Major Byard, a practice which did not follow the traditional text book strategy of developing a system that was discussed in Chapter Two. However, this prototyping process worked extremely well in the environment of developing the end-user system in house. Likewise, periodic reviews by the program manager enhanced the process even though (as mentioned above) problems arose as a result of the identification of new requirements.

Limitations

The fact that a single person performed all of the design work enabled that person to enjoy a steep learning curve, although the exact rate of learning was not measured. For instance, during the early design stages, it often took several hours to design a report. However, after Major Byard, the designer, became familiar with the special features of the software, he was able to generate new reports in a matter of minutes. Using a single, primary developer allowed for fast development of the system, and also resulted in an end product which reflected only a single methodology. On the other hand, the use of one developer leaves the organization with a very narrow base of "corporate knowledge" of the system. If the primary designer isn't present and available, there may be no one else with sufficient insight into the system to solve problems and make changes.

In addition, it is important to review the potential limitations of end-user development that were discussed in chapter two. For instance, standards, documentation, controls, testing, and interfaces with other systems, are often neglected, a fact which increases the risk to an organization. Also, users developing their own systems may underestimate the probability of errors. Knowing this, these areas were reviewed to ensure the system met the organization's goals.

Conclusion

All in all, the end-user strategy enabled the EM SPO to implement a financial management database system quickly without using a lot of resources. Even though the system has not undergone testing for an extensive amount of time, creative tests were developed; such as having four different users simultaneously access and make changes

to the same report in order to test the database's capabilities. Tests such as this demonstrated that the system is fully capable of meeting the needs and requirements of the SPO.

V. Conclusion

Introduction

In December of 1995, the Environmental Management (EM) SPO determined that it had a need for a financial management database system. It was determined that the resources existed that would allow for the system to be developed in house, by the actual end-users of the system. Potential risks of developing an end-user system existed, such as incomplete requirements analysis, bad design, and insufficient testing; however, an urgent need and the benefits of not requiring additional resources outweighed the risks. Even though the design and implementation of the financial management database system worked out extremely well for the EM SPO, it is not a project to be taken lightly.

Overview of Investigative Questions

The investigative question of determining the data requirements of ASC/FM was solved by creating PMR charts as part of the output of the system (Appendix B). The question concerning the information required by program managers was solved by creating the financial Checkbook as the other main output of the system (Appendix B). The question of whether or not the system could be designed in six months was answered, since the system was implemented in mid-June of 1996. Technically, the six month goal was not met, since the original starting date was early December of 1995.

However, the cause of this delay could be attributed to additional requirements incorporated into the system.

For instance, the information requirements analysis phase was more difficult than originally expected, due to the fact that top management did not force future users to decide on their requirements in a timely manner. This resulted in requirements being changed throughout the design process which resulted in delays of the final design and implementation of the financial database system. In addition, the possible delay of a new IFTS system resulted in changing one of the main requirements of the system. Eliminating the requirement of remaining parallel to IFTS, led to a decision to design a new database. This change resulted in a better overall system design, but also necessitated the creation of input screens, a feature not previously required.

Recommendations

Extensive research on the process of determining information requirements and my experience with observing and documenting the process followed in developing the EM SPO's financial management database system have convinced me that defining information requirements is the most important step in the development process. To be sure, the ultimate effectiveness of the system depends on the accuracy with which the information requirements are determined initially. Therefore, I recommend that future projects of the magnitude of this one begin with the establishment of a firm baseline for information requirements and that, except for absolutely unavoidable deviations, all development effort be accomplished in accordance with that baseline. In instances where

modifications to the system become necessary, proper consideration should be given to both the performance of the system and the effects on the schedule.

Future Research

Areas of future study include: evaluating the success of the system, especially the input screens, to analyze the accuracy of the database. In addition, it would be interesting to document any requested requirements changes in the future. For instance, expansion of the system to integrate schedule effects, such as linking schedule and budget and automatically analyzing manager and contractor execution performance. Another idea for useful research would be to use the database of environmental cost data to create a parametric cost model for predicting future environmental clean-up costs.

Appendix A: Data Dictionary

The following table provides the name of the field, and the type of field. The type of fields are date, character, and numeric. The table also gives the width of the field and the number of decimals in the numeric fields.

TABLE 3
DATA DICTIONARY FOR USER INPUT SCREENS

Name	Type	Width	Decimal
Project_no	Character	20	
Fy	Character	4	
Fund_oblig	Numeric	12	2
Oblig_date	Date	8	
Obfc_amt	Numeric	12	2
Obfc_date	Date	8	
Rev_date	Date	8	
Appn	Character	12	
Opr_name	Character	30	
Ocr_name	Character	30	
Title	Character	50	
Bp	Character	10	
Doc_type	Character	10	
Doc_no	Character	20	
Contr_no	Character	16	
Cont_mod	Character	8	
Descrip	Character	35	
Fund_req	Numeric	12	2
Fund_appr	Numeric	12	2
Appr_date	Date	8	
Fund_ba	Numeric	12	2
Ba_date	Date	8	
Fund_Rele	Numeric	12	2
Rele_date	Date	8	
Fund_commt	Numeric	12	2
Commt_date	Date	8	
Fund_spobl	Numeric	12	2
Spoobl_date	Date	8	
Fund_spoex	Numeric	12	2
Spoex_date	Date	8	
Fund_expen	Numeric	12	2
Expen_date	Date	8	

Name	Type	Width	Decimal
Expfc_amt	Numeric	12	2
Expfcc_date	Date	8	
Revex_date	Date	8	
Caward	Date	8	
Contractor	Character	20	
Mpc	Character	10	

Appendix B:

The Financial Management Database System User's Manual

Getting Started

Simply click on the financial weasel icon (the fox) in Microsoft Office. The first screen is the Financial Data Menu (Figure 5).

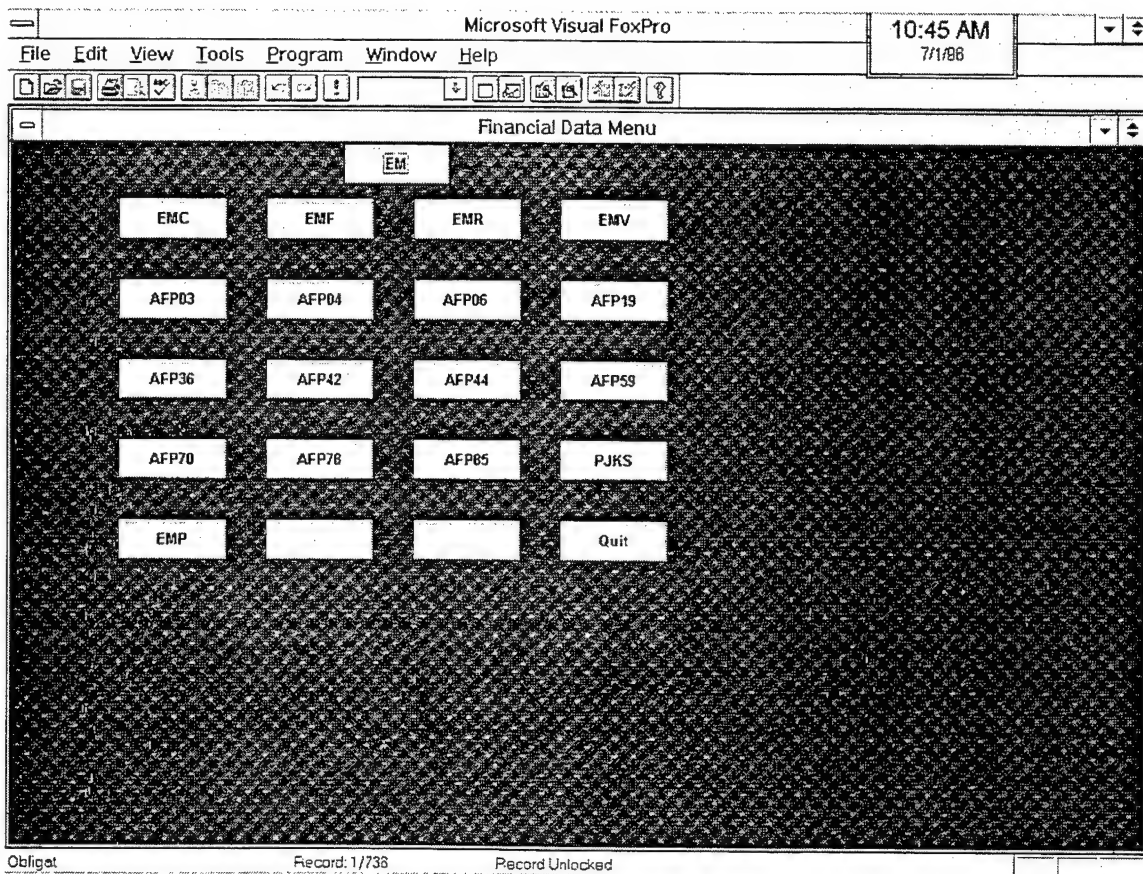


Figure 5. Financial Data Menu

The Financial Data Menu lists the four divisions of the Environmental Management (EM) SPO in the top row. Following the four divisions are twelve more buttons. The eleven buttons with AFP designators represent the active Air Force plants

being manage by the EM SPO. The PJKS button provides a list of the projects being managed. The last button of importance is the EMP button. This button allows Financial Management personnel to access the data input screens.

How To View A Particular Division's Data

Click on the button of the division you want to access. For example, clicking on the EMC button activates the Division Financial Data Menu. The EMC Financial Data Menu (Figure 6) is divided into fiscal years listed across the top of the screen. In addition, down the left side of the screen are the following categories: Checkbook, PMR Charts, and Project List.

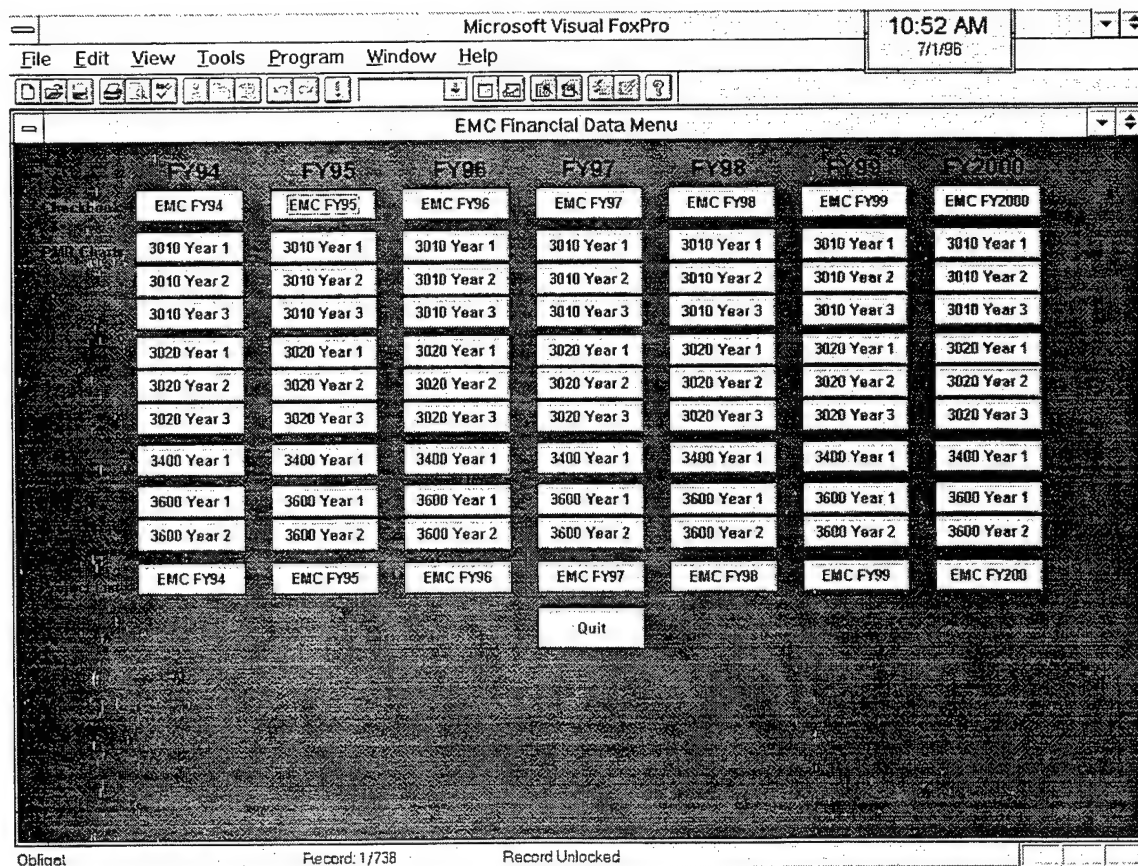


Figure 6. EMC Division Financial Data Menu

The Checkbook, also known as the Program Financial Statements, (Figure 3 or Figure 7) reflects funds required, approved, received, committed, obligated, expended, and unobligated, as well as percentages of the latter four in relation to the amount approved. These reports are structured as columns of data with rows for each project. The fact that these funds must be divisible by Division, IPT, Appropriation, and Fiscal Year, with summary lines for each provides the ability to cross-reference totals to ensure accuracy. One important feature to point out is that the menu is designed to be able to look at the status of a particular fiscal year by itself for each individual division.

The PMR Charts, also known as Funds Obligation Graphs, (Figure 4 or Figure 8) depict data graphically, with lines for cumulative funds obligated, forecast obligations, revised forecast obligations, and Office of the Secretary of Defense (OSD) obligation goals. Tabular summaries of these figures will also appear on this report. A few important notes include the fact that the various color of monies are split into different categories, while the number of years varies. This is due to the fact that 3010 and 3020 monies are for production and are designed to be obligated within three years. The system is then designed to track FY 95 3010 money in the EMC FY95 column under 3010 Year 1, as well as in the EMC FY96 column under 3010 Year 2, and in the EMC FY97 column under 3010 Year 3. This allows you to track and evaluate the original forecasted amounts, which are a major part of Program Management Reviews (PMRs).

The last category listed on the right hand side in Figure 6 is the Project List button. This button will provide a screen listing all of the active and inactive projects for a fiscal year in a particular division.

EMC Division Examples of Output

Now that the various categories have been explained, by pressing the EMC FY95 button in the Checkbook row of the EMC Financial Data Menu, an example of the EMC Division's Checkbook (Program Financial Statements) is displayed in Figure 7. (Note this is the same as Figure 3).

Report Designer - chkbopr.frx - Page 2 - Microsoft Visual FoxPro 10:57 AM
7/1/88

File Edit View Tools Program Window Help

Project Number	Project Title	MPC	Required	Approved	Committed	Obligated	Expended	Unobligated
EMC 1995								
0701/86 3010								
AFP 42								
19542301	BIO-REMEDATE FORMER UST SITE PH 22	7000	400,000.00	350,000.00	40,000.00	40,000.00	0.00	310,000.00
Totals			400,000.00	350,000.00	40,000.00	40,000.00	0.00	310,000.00
% of Approved Program					11.43 %	11.43 %	0.00 %	88.57 %
AFP 42L								
3945077	REMOVE UNDERGROUND FUEL UNE BLDG 215	7000	15,469.00	15,469.00	15,469.00	0.00	0.00	15,469.00
39542101	REM & REPL PCB BLDG 211&212-REMEDI SITE 2	7000	0.00	161,306.00	161,306.00	0.00	0.00	161,306.00
Totals			15,469.00	176,775.00	176,775.00	0.00	0.00	176,775.00
% of Approved Program					100.00 %	0.00 %	0.00 %	100.00 %
AFP 42N								
19400319	ASBESTOS & PCB ABATEMENT, B/301	7000	0.00	735,000.00	735,000.00	735,000.00	0.00	0.00
19542408	REMOV HYDRA LIFT/CONTAM ASPHALT BLD	7000	350,000.00	200,400.00	200,400.00	200,400.00	0.00	0.00
19542409	ASBESTOS ABATEMENT, SITE 3	7000	1,000,000.00	256,400.00	256,400.00	256,400.00	0.00	0.00
Totals			1,350,000.00	1,191,800.00	1,191,800.00	1,191,800.00	0.00	0.00
% of Approved Program					100.00 %	100.00 %	0.00 %	0.00 %
AFP 42P								
19542304	CNCT AFP 42P TO LANCASTER REGION WWTP	7000	2,719,000.00	2,719,000.00	2,719,000.00	2,719,000.00	0.00	0.00
39542307	QTRLY GW MONITORING & FIRE TRNG PIT	7000	95,000.00	95,000.00	95,000.00	95,000.00	0.00	0.00
Totals			2,814,000.00	2,814,000.00	2,814,000.00	2,814,000.00	0.00	0.00
% of Approved Program					100.00 %	100.00 %	0.00 %	0.00 %
AFP 42R								
19542201	ASBESTOS ABATEMENT, GDS 1 & 2, SITE 1	7000	350,000.00	350,000.00	350,000.00	0.00	0.00	350,000.00
39542305	RISK MANAGEMENT & PREVENTION PLAN	7000	0.00	30,000.00	30,000.00	30,000.00	0.00	0.00
Totals			350,000.00	380,000.00	380,000.00	30,000.00	0.00	350,000.00
% of Approved Program					100.00 %	7.89 %	0.00 %	92.11 %

Cu_checkbook Record: 35/81 Exclusive

Figure 7. Checkbook (Program Financial Statements)

Similarly, by pressing the EMC 3010 Year 1 button in the PMR Charts row of the EMC Financial Data Menu, an example of the the EMC Division's PMR Charts (Funds Obligation Graphs) is displayed in Figure 8. (Note this is the same as Figure 4).

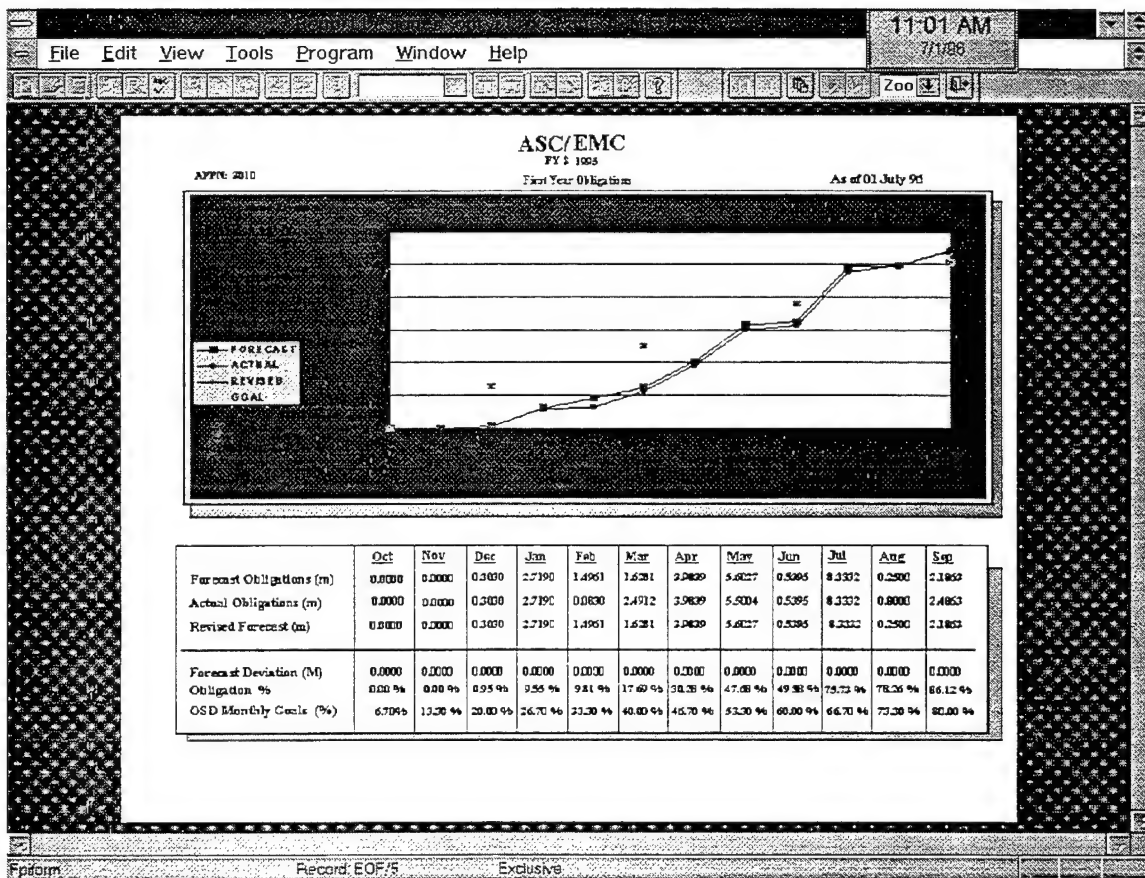


Figure 8. PMR Chart (Funds Obligation Graphs)

The above chart displays FY95 3010 obligations for the EMC Division. In particular, the graph displays the forecasted obligations, the actual obligations, and a revised forecast.

Likewise, by simply clicking on the appropriate button in the project list row from the EMC Financial Data Menu, a project list is displayed. For example, by clicking on the EMC FY95 button the following project list is displayed (Figure 9).

Year	Project Number	Project Name	Budget	Actual	Variance
1995	38510	LEAD ABATEMENT OF FOUNDRY BLDG	\$165,872.00	\$165,872.00	0.00
1995	232230	INSTALL CENTRAL COOLING TOWER	\$7,963,822.00	\$7,963,822.00	0.00
1995	295405	PAINT BOOTH EMISSIONS REDUCTION	\$354,283.00	\$354,283.00	0.00
1995	295408	BLDG 13 EMISSIONS CONTROL PH III	\$0.00	\$0.00	0.00
1995	295409	STORM WATER POLLUT PREVENTION STRUCTURES PH III	\$52,821.00	\$52,821.00	0.00
1995	295410	NPDES MONITORING SYSTEMS	\$145,471.00	\$145,471.00	0.00
1995	295411	PCB INVESTIGATION/REMEDATION	\$452,872.00	\$452,872.00	0.00

Figure 9. Project List

Figure 9 displays the FY95 project list for the EMC Division. Notice that the project number is given along with the name and dollar values. More importantly, it is important to notice the scroll bar on the far right hand side of the screen which provides the capability to scroll through the list.

Getting Back to the Menu

Each of the three previous screens provides you the opportunity to print the output. By pressing the escape button, you will be asked do you want to print or not. Regardless of your answer, the financial data menu will appear.

How to Use the Data Input Screens

At the main Financial Data Menu (Figure 5), click on the EMP button. The EMP Data Menu (Figure 10) is displayed.

The screenshot shows a Microsoft Visual FoxPro window titled "EMP Data Menu". The window contains a grid of buttons for data entry. The top row has buttons for "DERA FY94", "DERA FY95", "DERA FY96", and an empty button. Below these are three rows of empty buttons. The bottom row contains "Receive Funds", "Disburse Funds", an empty button, and "Quit". The status bar at the bottom shows "Cblfigst", "Record: 711/739", and "Record Unlocked".

Figure 10. The EMP Data Menu

At this time, only two buttons are of concern. They are the Receive Funds and the Disburse Funds buttons. By clicking on the Receive Funds Button, the Receiving Funds Input Screen is displayed (Figure 11).

Microsoft Visual FoxPro 10:41 AM 5/28/98

File Edit View Tools Program Window Help

Receive Funds

Receive Funds

Required	Approved	BA	Released	Committed	SPO Obligated	Obligated	Expenditure
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Project Number: 193577
 Fiscal Year: 1994
 Appropriation: 3010
 BPAC: 143000
 Title: REHAB VERT LIFT DOORS PHIV
 Description: REHAB VERT LIFT DOORS PHIV
 Opr Name: EMF
 OCR Name: AFP 04
 Mpc: 3000
 Fund ID:

Funds Required: 594,253.00
 Funds Approved: 706,177.00
 Funds BA: 706,177.00
 Approved Date: 11
 BA Date: 11

Forecast Obligation Amount: 706,177.00
 Forecast Obligation Date: 06/16/98
 Revised Date: 11
 Forecast Expenditure Amount:
 Forecast Expenditure Date:
 Revised Date: 11

Order By:

Project no	FY	Appn	OPR	OCR	Approved	Committed	Obligated
193577	1994	3010	EMF	AFP 04	706177.00	594253.00	594253.00
193577	1994	3010	EMF	AFP 04	0.00	111924.00	111924.00
193577SC	1994	SALES CREMF (SA	AFP 04		76000.00	76000.00	76000.00
193703	1994	3010	EMC	AFP 42L	171000.00	171000.00	171000.00
193705	1994	3010	EMC	AFP 42L	300000.00	300000.00	299993.88
193993	1994	3010	EMF	AFP 04	74773.00	74773.00	74773.00

Top Prev Next Bottom Find Print Add Edit Delete Cancel Save Exit

Skip to next record NUM CAPS

Figure 11. The Receive Funds Data Input Screen

This screen is used to input new projects and funds as they are received by the Directorate. The name of the fields available to the user along with a brief description are listed below:

Project Number: A unique project identifier

Fiscal Year:

Appropriation: The funding classification of the funds

BPAC:

Title: Title of associated project

Description: The description that would show up on a funding document in the disbursing screen

Opr Name: Highest level of management oversight

OCR Name: Plant number

Mpc:

Fund ID: Blank field that can be used to add unique identifier

Funds Required: Funds required in the Financial plan
Funds Approved: Funds approved for a particular project: can be different than financial plan
Funds BA: Actual Budget Authority received
Approved Date: Date notified of a new requirement
BA Date: Date of BA notification
Forecast Obligation Amount: Should equal amount of BA
Forecast Obligation Date: Date that obligation is projected to occur
Revised Date: If a change occurs, new date is added
Forecast Expenditure Amount: Amount of expenditures that will occur
Forecast Expenditure Date: Date that expenditure is projected to occur
Revised Date: If a change occurs, new date is added

The reason for having the obligation forecast fields accessible only from this screen is that we want to create a single line in the database reflecting the receipt of the money and the estimate of when it will be obligated. These values should appear only once and allowing access to these fields in subsequent transactions through the Disbursements screen risks double entries which would seriously corrupt the data. Each receipt of funds should be paired with the date it is projected to be obligated. In other words, on the very first entry for a project number, the forecast obligation amount will be entered and will serve as an anchor for that project number. Any additional entries dealing with the same project number will ignore the forecast obligation amount field in order to avoid double counting.

By hitting the escape button, you will be back at the EMP Data Menu. By clicking on the Disburse Funds button, the Disburse Funds Data Input Screen is displayed (Figure 12).

Microsoft Visual FoxPro 3:17 PM 5/30/88

File Edit View Tools Program Window Help

OBLIGAT

Commit, Obligate, Expend

193577 REHAB VERT LIFT DOORS PH IV

Required	Approved	BA	Released	Committed	SPO Obligated	Obligated	Expenditure
594,253.00	706,177.00	706,177.00	706,177.00	706,177.00	706,177.00	706,177.00	0.00

Project Number: 193577 SPO Expenditure: 0.00

FY: 1994 SPO Expenditure Date: 11

Appn: 3010 Expenditure: 0.00

BPAC: 140000 Expenditure Date: 11

Document Type: PR Contract Award Date: 09/23/94

Document Number: 94-0248 Contractor: LOCKHEED MARTIN

Contract Number: F39557-87-E-0193 Mpc: 3000

Contract Mod: P00059 Fund ID:

Description: REHAB VERT LIFT DOORS PH IV

Released: 594,253.00 SPO Expenditure:

Release Date: 11

Committed: 594,253.00

Commitment Date: 11

SPO Obligated: 594,253.00

SPO Obligation Date: 09/23/94

Obligated: 594,253.00

Obligation Date: 09/23/94

Project Number	FY	Appn	OCR	OPR	Document Type	Document Number	Committed	Obligated
193577	1994	3010	AFP	DEM	PR	94-0248	594,253.00	594,253.00
193577	1994	3010	AFP	DEM	PR	96-00010	111,924.00	111,924.00
193577SC	1994	SALES	CAF	DEM	TBD	TBD	76,000.00	76,000.00
193703	1994	3010	AFP	4EM	MIPR	94-5185	171,000.00	171,000.00
193705	1994	3010	AFP	4EM	PR	94-1008	300,000.00	299,993.00

Top Prev Next Bottom Goto Print Add Edit Delete Cancel Save Exit

Order By: Project Number Appropriation Fiscal Year OPR Name OCR Name

Skip to next record NUM

Figure 12. The Disburse Funds Data Input Screen

This screen is used to input all disbursements. The fields available to the user are listed below:

Project number
 FY:
 Appn:
 BPAC:
 Document Type:
 Document Number:
 Contract Number:
 Contract Mod:
 Description:
 Released:
 Released Date:
 Committed:
 Commitment Date:
 SPO Obligated:
 SPO Obligation Date:

Obligated:
Obligation Date:
SPO Expenditure:
SPO Expenditure Date:
Expenditure Date:
Contract Award Date:
Contractor:
Mpc:
Fund ID:

Bibliography

- Booch, Grady. Object Solutions: Managing the Object-Oriented Project. Menlo Park CA: Addison-Wesley Publishing Co., 1996.
- Bridger, Chet. "The Environment: Report Cites the High Price of Cleaning Up," Air Force Times, 42: 17 (30 October 1995).
- Byard, Major Kyle F. Chief of Program Control, ASC/EM, Wright-Patterson AFB OH. Personal Interviews. 1 September 1995 - May 1996.
- Courtney, James F. Jr. and David B Paradise. Database Systems for Management. (Second Edition). Homewood IL: Irwin, 1992.
- Davis, Gordon B. and Scott Hamilton. Managing Information: How Information Systems Impact Organizational Strategy. Burr Ridge IL: Irwin, 1993.
- El-Rewini, Hesham and Scott Hamilton. "Object Technology: A Virtual Roundtable," Computer, 28: 58,59 (October 1995).
- Headquarters Air Force Material Command. Financial Management Handbook. Book 2 of 2; attachment 3: 1443-1482. Wright-Patterson AFB OH.
- Hsia, Pei, Alan Davis and David Kung. "Status Report: Requirements Engineering," IEEE Software: 75-79 (November 1993).
- Kim, Young-Gul and Salvatore T. March. "Comparing Data Modeling Formalisms," Communications of the ACM, 38: 103-112 (June 1995).
- Lee, Byung S. "Object-Oriented Databases: Systems and Standards," Computer, 28: 64,65 (October 1995).
- Morse, Dave. "OODBSs and the Marriage of Network and Relational Database Models," Computer, 28: 66,67 (October 1995).
- Radding, Alan. "So What the Hell is OODBMS?," Computerworld, 29: 121- 129 (November 6, 1995).
- Sharp, Oliver. "Databases Get Objective," Byte, 20: 208DM 13 - 208DM 18 (November 1995).

Vessey, Iris and Sue A. Conger. "Requirements Specification: Learning Object, Process, and Data Methodologies," Communications of the ACM, 137: 102-113.
(May 1994).

Yadav, Surya B., Ralph R. Bravoco, Akemi T. Chatfield, and T.M. Rajkumar.
"Comparison of Analysis Techniques for Information Requirement
Determination," Communications of the ACM, 31: 1090-1099
(September 1988).

Vita

Captain Edward G. Leszynski was born on 26 June 1967 in Medina, Ohio. He graduated from Westlake High School in 1985 and entered the University of Notre Dame in South Bend, Indiana. He graduated with a Bachelors of Business Administration degree in Accounting and received his commission in May of 1989.

After completing the Accounting and Finance Officer Course at Shepherd AFB, his first assignment was to Kadena AB, Okinawa, Japan as a Deputy Accounting and Finance Officer. His next assignment was to Iraklion AS, Crete, Greece as an Accounting and Finance Officer. After serving two years as a Special Operation Forces Financial Manager at Wright-Patterson AFB, he entered the Graduate School of Logistics and Acquisition Management, Air Force Institute of Technology in 1995.

He has received his Master of Business Administration degree from Chapman University. In addition, he has passed the Certified Public Accountancy Exam and has completed Staff Officer School (SOS) by correspondence.

After receiving his Master of Science in Cost Analysis degree in September of 1996, he will be assigned to the Space and Missile System Center (SMC) at Los Angeles AS.

Permanent Address: 28250 Edgepark Blvd
 Westlake, Ohio 44145

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
<small>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.</small>				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE September 1996		3. REPORT TYPE AND DATES COVERED Master's Thesis
4. TITLE AND SUBTITLE THE DESIGN OF A FINANCIAL MANAGEMENT DATABASE SYSTEM			5. FUNDING NUMBERS	
6. AUTHOR(S) Edward G. Leszynski, Captain USAF				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology, WPAFB OH 45433-7765			8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GCA/LAS/96S-10	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) ASC/EM WPAFB OH 45433			10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) <p>This research led to the design and development of a financial management database system for the Aeronautical Systems Center (ASC) Environmental Management (EM) Systems Program Office (SPO), which has the responsibility of managing the environmental contracts for the Government-Owned, Contractor-Operated (GOCO) plants that are owned by the Air Force. This thesis investigated the various "development strategies" and "methodologies" described in the Management Information Systems literature in order to devise an end-user development strategy capable of meeting the EM SPO's requirements. In addition, the information requirements, conceptual design and prototyping, and procedures phases of the System Development Life Cycle (SDLC) are discussed in detail.</p> <p>Even though the objective of providing accurate and timely information to program managers was met, problems arose during the design and development phases as a result of changes in requirements. The author thinks that defining information requirements is the most important step in the development process because the ultimate effectiveness of the system depends on how accurately the information requirements are determined initially. Therefore, it is recommended that before a project of this magnitude is undertaken in the future, a firm baseline of system requirements be established early and be adhered to throughout the duration of the project.</p>				
14. SUBJECT TERMS Financial Management Database Information Requirements Analysis			15. NUMBER OF PAGES 76	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	

AFIT Control Number AFIT/GCA/LAS/96S-10

AFIT RESEARCH ASSESSMENT

The purpose of this questionnaire is to determine the potential for current and future applications of AFIT thesis research. **Please return completed questionnaire to:** AIR FORCE INSTITUTE OF TECHNOLOGY/LAC, 2950 P STREET, WRIGHT-PATTERSON AFB OH 45433-7765. Your response is **important**. Thank you.

1. Did this research contribute to a current research project? a. Yes b. No
2. Do you believe this research topic is significant enough that it would have been researched (or contracted) by your organization or another agency if AFIT had not researched it?
a. Yes b. No

3. **Please estimate** what this research would have cost in terms of manpower and dollars if it had been accomplished under contract or if it had been done in-house.

Man Years _____ \$ _____

4. Whether or not you were able to establish an equivalent value for this research (in Question 3), what is your estimate of its significance?

a. Highly b. Significant c. Slightly d. Of No
Significant Significant Significance

5. Comments (Please feel free to use a separate sheet for more detailed answers and include it with this form):

Name and Grade

Organization

Position or Title

Address